Topology aware adaptive security

Conference or Workshop Item

Topology aware adaptive security

Conference Item

Topology Aware Adaptive Security*

Liliana Pasquale[1], Carlo Ghezzi[2], Claudio Menghi[2], Christos Tsigkanos[2], Bashar Nuseibeh[1,3]
[1]Lero - the Irish Software Engineering Research Centre, University of Limerick, Ireland
[2]Politecnico di Milano, Milano, Italy
[3]The Open University, Milton Keynes, UK

ABSTRACT

Adaptive security systems aim to protect valuable assets in the face of changes in their operational environment. They do so by monitoring and analysing this environment, and deploying security functions that satisfy some protection (security, privacy, or forensic) requirements. In this paper, we suggest that a key characteristic for engineering adaptive security is the *topology* of the operational environment, which represents a physical and/or a digital space - including its structural relationships, such as containment, proximity, and reachability. For adaptive security, topology expresses a rich representation of context that can provide a system with both structural and semantic awareness of important contextual characteristics. These include the location of assets being protected or the proximity of potentially threatening agents that might harm them. Security-related actions, such as the physical movement of an actor from a room to another in a building, may be viewed as topological changes. The detection of a possible undesired topological change (such as an actor possessing a safe's key entering the room where the safe is located) may lead to the decision to deploy a particular security control to protect the relevant asset. This position paper advocates topology awareness for more effective engineering of adaptive security. By monitoring changes in topology at runtime one can identify new or changing threats and attacks, and deploy adequate security controls accordingly. The paper elaborates on the notion of topology and provides a vision and research agenda on its role for systematically engineering adaptive security systems.

Categories and Subject Descriptors

D.2.10 [**Software Engineering**]: Design; S.9.1 [**Security and Privacy**]: Software Security Engineering

General Terms

Security, Design

*We acknowledge SFI grant 10/CE/I1855 and ERC Advanced Grants (ASAP) no. 291652 and (SMScom) no. 227977.

SEAMS '14, June 2–3, 2014, Hyderabad, India

Keywords

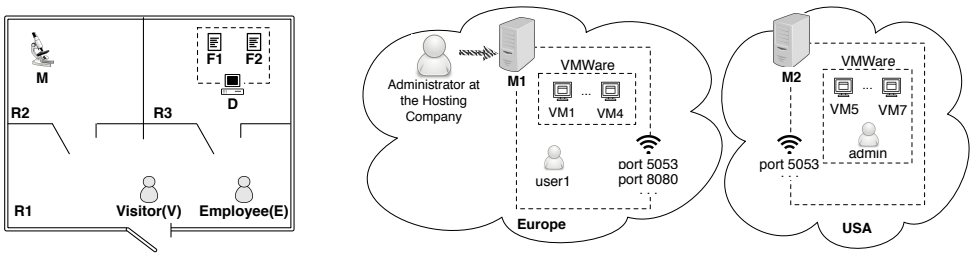
Topology, adaptation, security, privacy, digital forensics

1. INTRODUCTION

Engineering self-adaptive systems that continue to satisfy their security [17, 20], privacy [13] and forensic requirements [14] - hereafter collectively referred to as adaptive security systems - has been recently recognised as an important challenge to be addressed by the software engineering community. Adaptive security systems aim to protect valuable assets in the face of changes in their operational environment. Adaptive security is supported by monitoring and analysing the system operational environment, and deploying security controls that satisfy some protection (security, privacy, or forensic) requirements. Relevant characteristics of the operational environment of a system are invariably many and hard to determine and monitor. We suggest that a key characteristic is the topology of the operational environment - its structure in terms of the key elements and their relationships that determine the shape of the environment. In a physical sense, a topology denotes the physical characteristics of a space, such as size, adjacency, and connectivity, and is often represented as a map or physical model. In a digital sense, a topology often denotes structural characteristics of information, such as logical relationships between entities in an information model. In both cases, structural relationships are key, such as hierarchy, containment, proximity and reachability. Indeed, with the increasing expectations that systems are cyber physical, topology, we claim, can provide an important representation of context.

For adaptive security, topology can provide a system with both structural and semantic awareness of important contextual characteristics. These might include the location of assets being protected and the security controls that should be enacted in their close proximity. For example, a possible way to secure a sensitive document may be only to grant to a restricted set of trustworthy users permissions to access the room where the machine storing the document is placed. Moreover, the location of human and digital agents can also determine potential threats, which can harm the assets located in their vicinity. For example, the presence of a person in a room can represent a threat as she can steal any valuable asset placed in the areas reachable from her current location.

Changes in the topology due to movements of assets or agents can affect system security concerns, and indeed render existing security controls no longer effective. For example, the movement of a physical asset to a new location, may require applying stronger security controls on the area in which the asset is currently placed. Similarly, in a cloud infrastructure, the allocation of a new virtual machine (VM) used by a premium customer may require applying additional patches on the hosting physical machine. In these scenarios, a live representation of the topology at runtime can provide



(a) Physical topology (b) Digital topology

Figure 1: **Topology types.**

valuable contextual indicators as to where and when a system needs to deploy particular security controls to protect the relevant assets.

The role of topology [19] has been investigated in several domains that are not directly related to software engineering. In the wireless sensor networks community topology was not only used for sensor area placement, but also in the context of security. For example, adaptation in sensor network clustering in response to spam attacks has been proposed. However, existing context modelling techniques [4] do not explicitly capture the topology of the system operational environment. This position paper advocates topology awareness for more effective engineering of adaptive security. It clarifies the meaning of topology and its impact on the system security concerns and suggests that monitoring changes in topology and reasoning about their possible consequences at runtime can help identify new or changing threats and attacks, and deploy adequate security controls accordingly.

The rest of the paper is organised as follows. Section 2 introduces the notion of topology and explains how it can affect the identification of relevant security concerns. Sections 3, 4, 5 illustrate the research challenges topology brings for engineering adaptive security, privacy and digital forensics. Section 6 concludes the paper by providing future research questions to be addressed by the security engineering and the self-adaptive systems communities.

2. TOPOLOGY

This section provides a definition of topology and illustrates it with examples of physical and digital spaces. Finally, it explains how topology awareness can impact on the identification of security concerns such as assets, threats, attacks and vulnerabilities that are crucial for the selection of appropriate security controls.

2.1 Definition

Topology refers to the study of shapes and spaces, including properties such as connectedness and boundary [9]. A representation of the topology identifies the structure of space and the location of objects and agents in that space. A physical topology represents the location of physical agents (e.g., humans, robots) and objects in a physical environment (e.g., a building) and their structural relationships (e.g., agents-objects proximity). A digital topology represents the configuration of a virtual environment, such as a network, which may be composed of nodes (e.g., physical and virtual machines), including their hardware and software configuration and network connections. Topologies may include intangible areas such as administrative domains, where different security regulations apply. Both physical and digital topologies can also represent different kinds of relationships among the elements they represent such as containment, proximity, and reachability.

The proliferation of cyber physical systems is increasingly blurring the boundary between physical and digital topologies. On the one hand, digital objects, such as electronic files, can be stored on a desktop accessible in a physical space. On the other hand, physical objects, such as domestic appliances, can be accessed from digital objects, such as software applications. Even though architectural models for cyber physical systems have been proposed [3], they do not support reasoning on how changes in the topology can affect relevant security concerns and select security controls able to mitigate emerging security threats determined by such changes.

2.2 Example

Figure 1a shows a representation of the physical topology of a corporate building that is composed of rooms R1, R2 and R3. This topology also represents physical objects, such as lab equipment (e.g., microscope M) and a desktop (D) that are located in rooms R2 and R3, respectively, and human agents such as a visitor (V) and an employee (E). In this example, a *containment* relationship exists if an area contains objects/agents (e.g., room R1 contains agents V and E, or the building belongs to a specific department). A *proximity* relationship identifies the distance between two agents/objects or whether these are simply co-located in the same area. In this example, a visitor is co-located with an employee. A *reachability* relationship expresses if an agent can access another area or reach an object from a specific location. For example, room R2 can be accessed by the employee and the visitor who are in room R1, or M can be reached by those agents who are in room R2. For a physical topology, accessibility always requires agents-objects proximity.

Figure 1b shows a representation the digital topology of a virtualised network infrastructure. Objects can be physical machines (M1, M2), virtual machines (VM1, ... VM7), available or established network connections, applications and processes installed on physical and virtual machines, and files stored on these machines. Agents here represent users who are locally or remotely logged on the physical and virtual machines. A *containment* relationship represents applications/files installed/stored on physical machines, or can delimit the countries in which physical machines are located. For example, M1 and M2 are located respectively in Europe and in the USA. Containment relationships can also represent the fact that a user is running a particular application. For example the admin on M2 is running VMWare. The concept of *proximity* here represents the fact that different virtual machines, such as VM1, ..., VM4, are hosted on the same physical machine. Finally, *reachability* denotes the fact that an agent can access a physical or a virtual machine, by establishing a local or remote connection, or that a user can run a specific application/process because she has the permissions to do so. For example, the admin on M2 can access M1 because M1 accepts incoming network connections from M2, or the admin at the hosting company can locally access M1 - if she is physically co-located with M1. Moreover some of the applications/files installed/stored

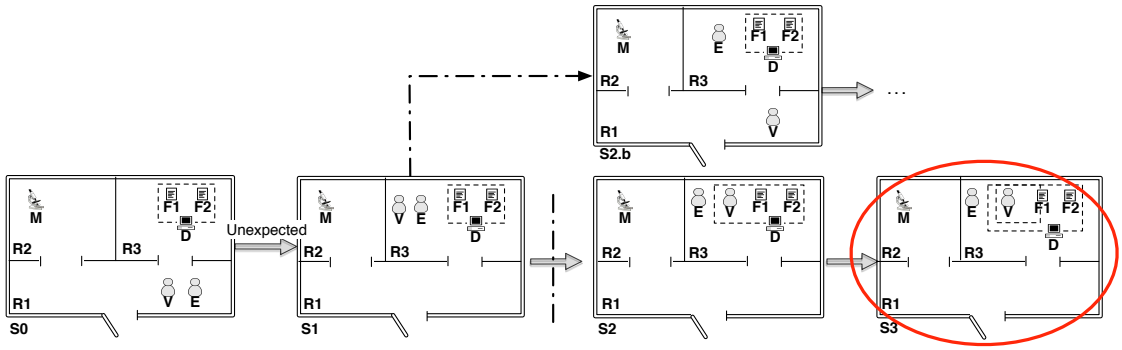


Figure 2: **Adaptive security scenario for the physical topology example.**

on M1 are accessible to user1 if there exists the relevant permission to execute/access them.

Note that there are a few differences between physical and digital topologies. Unlike physical spaces, in digital spaces reachability does not always require proximity because, for example, a machine can also be accessed remotely. Furthermore, while in physical topologies containments are well delimited, this is not always the case for digital topologies, where, for example, accessing a virtual machine does not require access to the physical machine hosting it. Finally, unlike physical topologies, in digital topologies an agent can reach different places at the same time. For example, the same agent can be in different processes or files, when it executes different processes or access different files at the same time.

2.3 Topology Awareness

Taking into account the topology of an operational environment can radically change the way we identify security concerns for engineering secure systems. Firstly, knowing where valuable assets are placed and their relationships with other objects in their proximity is crucial to identify possible security controls that can be enacted to protect them. For the physical topology shown in Figure 1a, security controls deal with identifying authentication and authorisation mechanisms to be put in place in some of the areas that need to be accessed to harm an asset. For example, to protect the microscope located in room R2, authentication and authorisation permissions for certain personnel must be applied to regulate access to R2. Likewise, digital assets such as F1 and F2, can be read or modified unlawfully depending on their accessibility in the physical space of the device on which they are stored (desktop D). In this case, possible security controls can regulate authorisation rights to read the document on D, or authentication and authorisation permission to log on to D or to access room R3 where D is located. Security controls can also enforce the relocation of assets to more secure areas. For example, M can be moved to another area (e.g., room R3) where access is restricted to trusted people.

Other security concerns, such as vulnerabilities, threats and attacks, can also depend on the locations of human and software agents, who can harm valuable assets placed in their vicinity. Vulnerabilities can be considered as capabilities offered by a physical or digital object, which can be exploited to harm an asset. The current topology state can give an indication of when a vulnerability can be exploited, for example, if an is agent is co-located with the same vulnerable object and has the capability to exploit it. Threats can arise from malicious agents, while attack vectors represent the possible sequences of actions that can be performed by an agent to harm an asset depending on the topology structure and relationships. For the example in Figure 1a, a threat may arise from a malicious visitor who can damage M by accessing room R2. Accessibility to room R2 from R1 is the exploited vulnerability. A visitor can also access confidential documents (e.g., F1) stored on D, if she enters in R3, logs on to C, and reads F1. Threat detection may lead to identifying possible security controls. For example, if a visitor inadvertently accesses an area where a valuable asset is located, she should be made to leave by a human or electronic guard.

For the digital topology shown in Figure 1b, assets to be protected can be virtual machines (e.g., those adopted by premium customers), as well as sensitive data that is stored and/or transmitted in the network. Threats and attacks directly depend on the configuration of physical and virtual machines from which the assets under protection can be reached. Threats can be malicious users, while potential attacks are determined by all possible ways in which an attacker can access the data or the VMs under protection by exploiting existing open ports and software vulnerabilities. For example, a threat scenario can revolve around an administrator at the hosting company, who can login on M1 locally and copy the image of the hosted virtual machines. Alternatively, a malicious attacker from M2 can exploit a vulnerability in the operating system installed on M1 to perform a buffer overflow attack targeting M1.

In this domain, security controls can modify access rights to data and VMs under protection, selectively apply patch updates to software installed on physical and virtual machines to fix vulnerabilities that can be exploited by potential attackers, dynamically modify firewall configurations, or forbid incoming network connections from specific hosts. Note that security controls can also be applied on machines that are not directly under protection and from which the assets to be protected are still reachable. For the example of Figure 1b, security controls can be applied on M1 to forbid incoming network connections from M2 or by patching the installed software to remove vulnerabilities. Alternatively, security controls can be applied to fix vulnerabilities in the software installed on M2, which can be exploited by an attacker to compromise M2 and remotely connect to M1. Finally, security controls can also enforce the relocation of virtual machines to more secure domains in which the foreseen attacks are no longer feasible.

3. ADAPTIVE SECURITY

Adaptive security [17] aims to continue to protect valuable assets from harm, even when security concerns change dynamically. To prevent potential attacks, security controls are adjusted at runtime depending on the varying risk of harm. In this section we discuss the implications that topology has on adaptive security by using the examples outlined in Section 2.2.

Topology changes can render an existing security configuration no longer effective. For a physical topology, moving a valuable

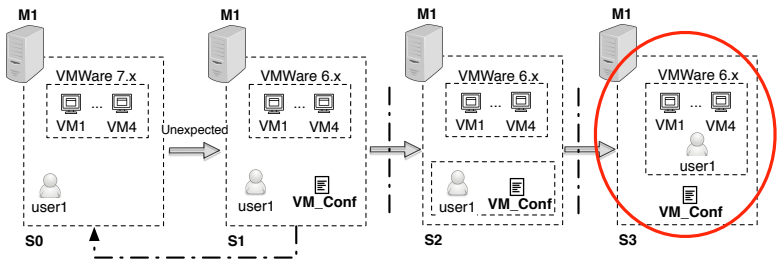


Figure 3: **Adaptive security scenario for the digital topology example.**

asset (e.g., microscope M) to a room (e.g., R3) may require enabling stronger access control mechanisms to the rooms/areas in which the asset is enclosed. Additionally, unexpected agents' movements may add new threats, as they can harm the assets that can be reached from their current location, which must be mitigated. For example, if a visitor accesses room R3 because she is accompanied by an authorised employee (see Figure 2, state S1) the authentication and authorisation mechanisms used to protect the assets in R3 may no longer be effective because in a future state (S3) a visitor can log on to desktop D and access the confidential files stored in it (F1).

To discover potential attacks caused by topological changes, it is necessary to reason about the actions that potentially malicious agents can perform to harm valuable assets, by exploiting the configuration of a physical space. For example, Figure 2 represents an attack scenario that can take place from state S1, when a visitor enters in room R3 together with an authorised employee. Note that each state transition is determined by an action changing the topology's structure and relationships (e.g., agents/assets movements, opened files). Different security controls can be identified (see dashed lines in Figure 2) by exploring the state space of the possible topology configurations. A possible solution is to prevent the system from reaching state S2 (i.e. a visitor accesses desktop D), for example, by applying stronger authentication mechanisms on D. Another solution is to force the system to reach a state from which the state associated with a security breach is more distant - in terms of number of state transitions. For example, a security control can force the vistor to leave room R3 (see state S2.b).

For a digital topology, the allocation of a new VM to a potentially malicious customer may cause harm to other customers who use VMs co-hosted on the same physical machine because covert channel attacks can be attempted. Changes in the network and software configuration of physical and virtual machines can add new vulnerabilities, which can be exploited by existing known attack modules. For example, if a system administrator at the hosting company downgrades the VMWare version installed on M1 to version 6.x for licensing reasons (see Figure 3, state S1), an existing vulnerability[1] allows a local user to gain root privileges by setting a library path option in a configuration file. Consequently, this user will be authorised to make copies of a target VM image hosted locally (state S3). Potential security controls can be identified by preventing a system from moving to state S2, by revoking the permission to modify the VMWare configuration file causing the vulnerability. Another possible option could be to force the system to rollback to a previous state, for example, by forcing the administrator to upgrade VMWare to a version greater than 6.x.

These scenarios require explicit support of "topology-awareness" in the activities of the MAPE (Monitoring, Analysis, Planning, Executing) loop [10] necessary for adaptive security, and topology should be conceived as a live entity at runtime. Topology-relevant changes, such as movements of software and human agents or VM re-locations, must be monitored and used to estimate the impact on related security concerns, such as potential threats and attacks and applicable security controls. Analysis techniques, similar to those intuitively explained in Figure 2 and Figure 3 must be provided to reason about potentially harmful scenarios that are feasible in the future states of a system and can harm a set of assets under protection. Similarly, planning techniques should allow identifying actions that prevent the system to move to an harmful state, or force rollback to a state in which the harmful scenario is no loger plausible. Finally the effects of the execution phase, modifying the topology or restricting the available movements that agents can perform, must also be synchronised with the current representation of the topology.

Fuzzy causal networks have been recently adopted in previous work [17] to adaptively re-estimate security risk when asset-relevant changes take place, and to identify a suitable configuration of security controls to apply at runtime. However, previous work does not take into account topological changes as a trigger for risk re-estimation and adaptation. To overcome this limitation, formalisms such as spatial logics [1] can be used to express properties on the topological conformation of the operational environment in order to formally define harmful states that a system must never reach. In addition, existing calculi such as pi-calculus [12] or ambient calculus [8], can be employed to reason about a set of potential states that a system can reach to mitigate existing threats. As far as we are aware, existing model checking techniques [5, 6, 7] that use these formalisms have only been employed to verify security policies but they have not been adopted to assess security risks or to suggest possible security controls that can be applied in specific situations.

Another challenge to support adaptive security is the need to perform analysis and planning activities efficiently at runtime. To deal with the complexity of risk assessment for large and distributed topologies, such as those representing cloud infrastructures or smart cities, it is possible to reduce the analysis space by using the notion of topology itself. Containment relationships allow identifying only specific areas of the problem space in which the reasoning must be performed. For example, the reasoning could focus only on potential threats that can harm assets located in the areas affected by a change. Another possibility is to only perform the reasoning up to a maximum number of subsequent actions (e.g., movements in a physical topology or execution of attack modules in a virtual topology) attackers can use to harm assets under protection. A combination of these strategies could also be used.

4. ADAPTIVE PRIVACY

Adaptive privacy [13] aims to continue to protect personal and sensitive information from unauthorised collection, storage, use, and transmission. To prevent potential privacy breaches, suitable actions that regulate the level of information disclosed are suggested to a

[1]http://www.cvedetails.com/cve/ CVE-2008-0967/

user at runtime depending on the varying risk of harm and the social benefits deriving from the disclosure.

In the privacy domain, the main asset to be protected is personal information, which may be stored on digital files (e.g., text files, pictures) or can be inferred from the users' interaction with digital devices (e.g., sensors tracking users' location or movements). Privacy threats cause harm to an individual or to a group of people who are related to the content of the information disclosed. Therefore, estimating privacy risks depends on the negative consequences that disclosing information can have on its subjects. Social benefits deriving from the disclosure also need to be traded-off with privacy risks in order to provide appropriate suggestions on whether to selectively disclose a specific piece of information.

Taking into account the topology of the operational environment can radically change the way we identify privacy concerns. Firstly, topology can have a direct impact on the sensitivity of an information item. For example, location information, which at a first glance is not perceived as sensitive, may topologically be related to additional information that actually is very sensitive. For example, location information implying proximity to potentially embarrassing places (e.g., nightclubs) is more sensitive than location information revealing proximity to a person's workplace. Topology can also determine privacy threats. In particular, agents co-located in the same physical space with an individual may represent a threat because they can potentially intercept the information she is disclosing. For example, a threat scenario may revolve around a person providing as input to her mobile device the password of her bank account in close proximity to unknown people. Furthermore, the recipients of an information item transmitted by a person can determine a threat scenario if one of them re-transmits this information to some of her contacts. Another threat scenario may revolve around a person sharing a sensitive picture with her social network friends; in case one of her friends in turn shares the picture with friends of friends, this may lead to a privacy violation.

Topology changes can affect sensitivity of information and bring new threats, rendering existing disclosure interactions more or less harmful. A person's movements can affect sensitivity of disclosed information. For example, location information can become more sensitive when a person moves to specific locations (e.g., to a hospital, because it can reveal an individual's health issues). Additionally, privacy threats can arise from agents' movements. For example, if an agent comes in close proximity to a person when she is providing the password of her bank account, this password has a higher chance to be intercepted. Changes in the connections topology of a person's social network contact can also bring new threats. For example, consider the scenario in which a person's contact (Alice's contact Bob) adds to the network of his friends another contact (Charlie, who is Alice's work colleague). In this case, when Alice shares some of her personal information to her social network contacts potential threats can arise. This is due to the fact that shared information can potentially be received by undesired recipients (Charlie), because Bob can share Alices' information with his contacts.

All these scenarios require explicit support of "topology-awareness" in the activities of the MAPE loop to support adaptive privacy. Monitoring must detect topological changes at runtime to re-estimate privacy concerns, such as sensitivity of disclosed information and new/changing agents behaviours enabled by topological changes. Potential solutions could benefit from existing studies in ubiquitous computing and human factors [11] to better understand privacy concerns of users sharing information in relation to their topological context. Analysis may be triggered by acts of information disclosure to assess its potential threats on users' privacy. Existing work on adaptive privacy [13] detects privacy threats by checking violation of privacy requirements over behavioural models of system agents. However, we are not aware of any work to identify variations in agents' potential behaviours that can be determined by changes in topology. Such work could help understand how topological changes can dynamically bring new/different privacy breaches. Planning must identify different levels of information disclosure that a user can enable, and associate each of them with potential threats (e.g., which recipients does a specific information disclosure reach). This will guide users to decide how to store, insert, and transmit their personal information. Execution differs from its security counterpart since it is mainly the responsibility of the user to select and enact a specific level of information disclosure depending on the suggestions given by the planning.

Existing privacy laws and standards regulating the treatment of personal information by external parties also depend on where information resides. For the example shown in Figure 1b, physical machines M1 and M2 are located in Europe and in the USA respectively, where completely different privacy regulations apply. The European Data Protection Directive[2] grants users the right to both remove and correct any personal information online. In contrast, in the USA customers do not have the same rights of controlling their own information online, except for medical records [2]. Thus, location of information determines the actions that are permitted by the law for the handling of personal data, and topology awareness can therefore help identify privacy breaches if regulations being in force do not comply with citizens' privacy requirements.

Topology awareness in adaptive privacy can also provide tangible benefits in maintaining privacy requirements when customers outsource part of their services to the cloud. Topology changes, such as VMs re-location or data replication, may cause movements of information to a different jurisdiction where users' privacy requirements may no longer be satisfied. In this scenario, topology awareness may help support the novel notion of *adaptive compliance*, which aim to identify changing jurisdictions and apply adequate adaptation actions to maximise the satisfaction of customer' privacy requirements. To support adaptive compliance, it is necessary to identify formal techniques (e.g., deontic logic [18]) to represent customers' privacy requirements and privacy regulations being in force in different jurisdictions. Monitoring should support data provenance and track the jurisdictions where information resides. Analysis should use verification techniques (e.g., [15]) to check customers' privacy requirements. If some of these requirements cannot be satisfied, planning should identify suitable adaptation strategies (e.g., VMs and storage re-location), and, if no strategy is suitable, customers must be notified of the privacy breaches and have the possibility to re-negotiate their contract with the service provider.

5. ADAPTIVE DIGITAL FORENSICS

A software engineering challenge in digital forensics is to build systems that are forensic-ready [16], which are able to support the potential collection and use of digital evidence. To assess how a crime (i.e. a security breach) was perpetrated, such systems must perform targeted evidence collection, perhaps even before a crime takes place. This is fundamental especially when evidence is volatile, as it may no longer be available subsequently. Indeed, evidence can be concealed by potential attackers or can come from volatile sources (e.g. volatile storage). Evidence ephemerality is also a typical problem of cloud infrastructures, which provide an 'elastic' environment, where storage and computing resources are provided and released on demand. Therefore evidence can be lost if it is not preserved adequately.

[2]Directive 95/46/EC

Collecting all possible evidence proactively is not a viable solution, since it can be very voluminous and cumbersome to analyse. Instead, evidence collection activities must only focus on gathering the data necessary to investigate potential attacks that can exploit the current system configuration. In this scenario, adaptive digital forensics [14] aims to continue to support forensic-readiness even when potential attacks change due to modifications of the external operational environment. As demonstrated in section 3, topology changes can modify the actions that can be performed by an attacker to generate security breaches. Moreover, knowing potential attacks in advance can allow us to focus evidence collection activities only on those assets and locations exploited by an attacker, and avoid collecting evidence from all possible sources.

We suggest that topology awareness should be considered for engineering the activities of the MAPE loop to support adaptive digital forensics. As with adaptive security, monitoring and analysis must track topology-relevant changes, such as movements of software and human agents or VMs re-location, and estimate their impact on related security concerns, such as potential threats and attacks. For each potential attack, the planning activity must identify suitable strategies to preserve the evidence coming from the digital objects whose vulnerabilities are exploited to perpetrate an attack. Execution must enable evidence collection activities identified during planning, and disable those that are no longer necessary, because an attack is no longer feasible. For the scenario shown in Figure 2, when a system unexpectedly moves to state S1, evidence collection activities will monitor users' logins on desktop D and read/write/copy/delete operations performed on file F1, whose integrity must be preserved. For the scenario shown in Figure 3, if a system moves unexpectedly to state S1, accesses of a user to a VMWare configuration file (VM_Conf) that can be manipulated to perform an attack must be monitored together with all VMWare operations aimed to perform copies of the VM images hosted locally.

6. AN EMERGING RESEARCH AGENDA

This position paper has provided our vision for the use of topology for engineering adaptive security systems, and demonstrated - by utilising examples of physical and digital spaces - how topology awareness can help engineer more effective adaptive security. We believe several research questions emerged from our paper that the self-adaptive system community could address. Firstly, it is necessary to identify appropriate formalisms (e.g., [8, 12]) to represent topology and track its changes at runtime. Secondly, analysis techniques must be further investigated to understand how topological changes affect security and privacy concerns. To achieve this aim, model checking of spatial properties could be a suitable way to identify potential attacks that can exploit the topology of the operational environment. Moreover, planning techniques are needed to generate adaptation actions that can prevent or mitigate security and privacy breaches determined by topological changes. In particular, planning could take into account adaptation costs (penalisation of other non-security requirements) and benefits (risk mitigation). Both planning and analysis can exploit the topology of the surrounding environment to reduce the state space, thus leading to more efficient adaptation at runtime. We hope both the security engineering and the self-adaptive systems communities will find the research questions and challenges highlighted in this paper useful in order to achieve more effective adaptive security.

7. REFERENCES

[1] M. Aiello, I. Pratt-Hartmann, and J. Van Benthem. What is Spatial Logic? In *Handbook of Spatial Logics*, pages 1–11. Springer, 2007.

[2] A. I. Antón, J. B. Earp, M. W. Vail, N. Jain, C. M. Gheen, and J. M. Frink. HIPAA's Effect on Web Site Privacy Policies. *IEEE Security & Privacy*, 5(1), 2007.

[3] A. Bhave, D. Garlan, B. Krogh, A. Rajhans, and B. Schmerl. Augmenting Software Architectures with Physical Components. In *Proc. of the Embedded Real Time Software and Systems Conf.*, pages 19–21, 2010.

[4] C. Bolchini, C. A. Curino, E. Quintarelli, F. A. Schreiber, and L. Tanca. A Data-oriented Survey of Context Models. *SIGMOD Record*, 36(4):19–26, 2007.

[5] C. Braghin, A. Cortesi, and R. Focardi. Security Boundaries in Mobile Ambients. *Computer Languages, Systems & Structures*, 28(1):101–127, 2002.

[6] C. Braghin, A. Cortesi, and R. Focardi. Information Flow Security in Boundary Ambients. *Information and Computation*, 206(2):460–489, 2008.

[7] C. Braghin, N. Sharygina, and K. Barone-Adesi. A Model Checking-Based Approach for Security Policy Verification of Mobile Systems. *Formal Aspects of Computing*, 23(5):627–648, 2011.

[8] L. Cardelli and A. D. Gordon. Mobile ambients. *Theor. Comput. Sci.*, 240(1):177–213, 2000.

[9] L. Euler. Solutio Problematis ad Geometriam Situs Pertinentis. *Commentarii Academiae Scientiarum Petropolitanae*, 8:128–140, 1741.

[10] J. O. Kephart and D. M. Chess. The Vision of Autonomic Computing. *IEEE Computer*, 36(1):41–50, 2003.

[11] C. Mancini, K. Thomas, Y. Rogers, B. A. Price, L. Jedrzejczyk, A. K. Bandara, A. N. Joinson, and B. Nuseibeh. From Spaces to Places: Emerging Contexts in Mobile Privacy. In *Proc. of the 11th Int. Conf. on Ubiquitous Computing*, pages 1–10, 2009.

[12] R. Milner. *Communicating Systems and The Calculus*. Cambridge University Press, 2003.

[13] I. Omoronyia, L. Cavallaro, M. Salehie, L. Pasquale, and B. Nuseibeh. Engineering Adaptive Privacy: On the Role of Privacy Awareness Requirements. In *Proc. of the 35th Int. Conf. on Software Engineering*, pages 632–641, 2013.

[14] L. Pasquale, Y. Yu, M. Salehie, L. Cavallaro, T. T. Tun, and B. Nuseibeh. Requirements-Driven Adaptive Digital Forensics. In *Proc. of the 21st Int. Requirements Engineering Conf.*, pages 340–341, 2013.

[15] F. Raimondi and A. Lomuscio. Automatic Verification of Deontic Interpreted Systems by Model Checking via OBDD's. In *ECAI*, volume 16, page 53, 2004.

[16] R. Rowlingson. A Ten Step Process for Forensic Readiness. *International Journal of Digital Evidence*, 2(3):1–28, 2004.

[17] M. Salehie, L. Pasquale, I. Omoronyia, R. Ali, and B. Nuseibeh. Requirements-Driven Adaptive Security: Protecting Variable Assets at Runtime. In *Proc. of the 20th Int. Requirements Engineering Conf.*, pages 111–120, 2012.

[18] G. H. Von Wright. Deontic Logic. *Mind*, 60(237):1–15, 1951.

[19] J. Wang and G. M. Provan. A Comparative Analysis of Specific Spatial Network Topological Models. In *Proc. of the 1st Int. Conf. on Complex Sciences*, pages 1514–1525, 2009.

[20] E. Yuan and S. Malek. A Taxonomy and Survey of Self-Protecting Software Systems. In *Proc. of the 7th Int. Symp. on Software Engineering for Adaptive and Self-Managing Systems*, pages 109–118, 2012.